Towards a Usability Evaluation Process for Model-Driven Web Development

Adrian Fernandez
ISSI Research Group
Department of Information Systems and Computation - Universidad Politécnica de Valencia, Camino de Vera, s/n, 46022, Valencia, Spain.
+34 96 387 73 50
afernandez@dsic.upv.es

Emilio Insfran
ISSI Research Group
Department of Information Systems and Computation - Universidad Politécnica de Valencia, Camino de Vera, s/n, 46022, Valencia, Spain.
+34 96 387 73 50
einsfran@dsic.upv.es

Silvia Abrahão
ISSI Research Group
Department of Information Systems and Computation - Universidad Politécnica de Valencia, Camino de Vera, s/n, 46022, Valencia, Spain.
+34 96 387 73 50
sabrahao@dsic.upv.es

ABSTRACT
This paper presents an approach to integrate usability evaluations into Model-Driven Web development processes. Our main motivation is to define a generic usability evaluation process which can be instantiated into any concrete Web development process that follows a Model-Driven Development (MDD) approach. A preliminary version of a Web Usability Model was defined in order to support this usability evaluation process at several stages. This Web Usability Model decomposes the usability sub-characteristics (from the Software Quality Model proposed in the ISO/IEC 25000 SQuaRE standard) into other sub-characteristics and measurable attributes. Web metrics are intended to be associated to measurable attributes in order to quantify them. Our approach is intended to perform usability evaluations at several stages of a Web Development process. In this paper, we show how usability evaluations at final user interface (UI) can provide feedback about changes in order to improve usability issues at intermediate artifacts (Platform-Independent Models and Platform-Specific Models) or at transformations rules among these intermediate artifacts.

Categories and Subject Descriptors
D.2.9 [**Management**]: *Software quality assurance*, D.2.8 [**Metrics**]: *product metrics*. H5.2 [**User Interfaces**]: *Evaluation/methodology*

General Terms
Measurement, Design.

Keywords
Web Usability Model, Usability Evaluation, Web Metrics, Model-Driven Development.

1. INTRODUCTION
Usability in Web applications is a crucial factor since the ease or difficulty that users experience with this kind of systems will determine their success or failure. Web applications are increasing its importance in industrial domains; thereby, the need for usability evaluation methods that are specifically crafted for the Web domain has become critical.

I-USED '09, August 24, 2009, Upssala, Sweeden.

Usability evaluations methods for Web applications can be supported by a quality model which defines usability as a quality characteristic that is decomposed into specific attributes that are easier to measure. Although there are several proposes in this field, most of these approaches [12],[13] only consider usability evaluation at final stages when the product is almost completed where correcting its usability problems is more difficult. It is widely accepted that evaluations performed at each phase of Web applications development is a critical part of ensuring that the product will actually be used and be effective for its intended purpose. We argue that integrating usability issues into the MDD approach can be an effective way to reach this objective since the quality evaluation of intermediate artifacts (models that specify an entire Web application), is applied in all steps of the process [2]. A Web development process that follows a MDD approach basically transforms models that are independent from implementation details (Platform-Independent Models - PIM) into other models that contain specific aspects from a concrete platform (Platform-Specific Models - PSM). Transformation rules, which are applied at PSMs, are able to automatically generate the Web application source code (Code Model - CM).

This paper presents an approach to integrate usability evaluation into any Model-Driven Web Development method by defining a usability evaluation process. This Web Usability Model has been defined by decomposing the usability sub-characteristics (from the Software Quality Model proposed in the ISO/IEC 25000 SQuaRE standard) into other sub-characteristics and measurable attributes taking into account ergonomic criteria proposed in Bastien and Scapin [4]. Although our approach is intended to perform usability evaluations at several stages of a Web development process, in this paper, we mainly focus on how evaluations at final user interface (Code Model) can provide feedback about changes in order to improve usability issues at intermediate artifacts (PIM and PSM models) produced at early stages of the Web development process and at transformations rules among these intermediate artifacts.

This paper is organized as follows. Section 2 discusses related work that report usability evaluation processes for Web applications. Section 3 presents our approach to integrate usability evaluations into Model-Driven Web Development. Section 4 presents our Web Usability Model that supports our approach. Section 5 shows a brief example of how the usability evaluation process can be instantiated into a concrete Web development method. We mainly focus on evaluations at final user interface. Finally, Section 6 presents discussions and further work.

2. RELATED WORK

There are several approaches that deal with Web usability evaluation, for instance, Ivory [16], Olsina and Rossi [13], Calero *et al.* [5], Seffah *et al.* [15], and Moraga *et al.* [12].

Ivory [16] presents a methodology for evaluating information-centric Web sites. The methodology proposes five stages: identifying an exhaustive set of quantitative interface measures such as the amount of text on a page, color usage, consistency, etc; computing measures for a large sample of rated interfaces; deriving statistical models from the measures and ratings; using the models to predict ratings for new interfaces; and validating model prediction.

Olsina and Rossi [13] proposed the Web Quality Evaluation Method (WebQEM) to define an evaluation process in four technical phases: Quality requirements definition and specification (specifying characteristics and attributes based on the ISO/IEC 9126-1 [9]. such as *usability*, *functionality*, *reliability,* and *effectiveness* and taking into account Web audience's needs), elementary evaluation (applying metrics to quantify attributes), global evaluation (selecting aggregation criteria and a scoring model), and conclusion (giving recommendations). Nevertheless, evaluations take place mainly when the application is completed.

Calero *et al.* [5] present the Web Quality Model (WQM), which is intended to evaluate a Web application according to three dimensions: Web features (*content*, *presentation*, and *navigation*); quality characteristics based on the ISO/IEC 9126-1 (*functionality*, *reliability*, *usability*, *efficiency*, *portability*, and *maintainability*); and lifecycle processes (*development*, *operation* and *maintenance*) including organizational processes such as *project management* and reuse *program management*. WQM has been used to classify, according to these three dimensions, a total of 326 Web metrics taken from the existing literature. An evaluation process can be defined by selecting the most useful set of metrics to construct a "total Web quality" expression that could be used to quantify the quality of a given Web application. However, guidelines on how to define this process have not been provided.

Seffah *et al.* [15] present the Quality in Use Integrated Measurement (QUIM) as a consolidated model for usability measurement in Web applications. An editor tool has presented to define measurement plans collecting data from different combinations of metrics proposed in the model. QUIM combines existing models from ISO/IEC 9126-1 [9], ISO/IEC 9241-11 [8], and others. It decomposes usability into factors, and then into criteria. In this approach, a criterion can belong to different factors. Finally, these criteria are decomposed into specific metrics that can quantify the criteria.

Moraga *et al.* [12] present a usability model towards portlet evaluation. Portlets are pluggable user interface software components that are managed and displayed in a web portal. The portlet usability model is based on the sub-characteristics from ISO/IEC 9126 (understandability, learnability and compliance), nevertheless, the operability sub-characteristic was replaced by customizability which is closer to the portlet context. The usability evaluation process proposed is based on a number of ranking with acceptance thresholds in order to quantify the sub-characteristics from the models.

The majority of these approaches evaluate Web applications in order to suggest changes at design or implementation stages. It implies that more efforts and resources must be invested into code maintenance. This fact does not occur in a MDD approach where only the maintenance of models is required since source code can be automatically generated from the intermediate artifacts (PIM and PSM models).

In previous work, Abrahão and Insfran [3] proposed a usability model for early evaluation in model-driven architecture environments. Usability was decomposed into the same sub-characteristics as the ones in the ISO/IEC 9126 (*learnability*, *understandability*, *operability*, and *compliance*), and then decomposed again, into more detailed sub-characteristics and attributes. However, the model did not provide metrics for measuring the model attributes and it was not proposed specifically for the Web domain. Panach *et al.* [14] presents an adaptation from the previous model to the Web domain in order to evaluate usability at PIM models for a concrete and proprietary Model-Driven Web Development approach.

As far as we know, there is no proposal for a generic usability evaluation process supported by a usability model in the Model-Driven Web Development context.

3. THE USABILITY EVALUATION PROCESS

Since the adoption of Model-Driven Development (MDD) in the industrial domain has increased recently, our approach is intended to integrate usability issues into a Model-Driven Web Development. Web development methods that follow this approach such as OO-H [7], WebML [6], or UWE [11] support the development of a Web application by defining different views (models), including at least one structural model, a navigational model, and an abstract presentation model. These methods also provide model transformations and automatic code generation.

The usability of a Web application obtained as a result of a MDD process can be assessed at different abstraction levels (PIM, PSM and CM). Our proposal is intended to use a Web Usability Model, which is a set of sub-characteristics decomposed into measurable attributes that can be quantified by metrics. The Web Usability Model can be applied to assess the models from each abstraction level (see Fig.1). However, not all the measurable attributes can be evaluated at all the abstraction levels. The higher abstraction level, the less attributes can be considered. In addition, feedback that is obtained after the artifact evaluation has different targets depending on the abstraction level:

1. At the PIM level it is possible to assess models that specify the Web application independently of platform details such as: presentation models, navigational models, dialogue models, etc. (1 in Fig.1). The set of measurable attributes that can be evaluated at this level is mainly related to how the information will be accessed by users and how this information will be presented by abstract UI patterns (i.e. navigability, information density, etc). However, this set of attributes may differ depending on the PIM expressiveness from each Web development method. This evaluation will generate a usability report in order to provide feedback about how to correct these PIM models.
2. At the PSM level it is possible to assess the concrete interface models related to a specific platform (2 in Fig.1). The set of measurable attributes that can be evaluated at this level is wider since it includes attributes related with specific software components (widgets) that cannot be considered at PIM level (i.e. behavior of explore bars, visual feedback from radio buttons, etc).This evaluation will generate a usability report in

order to provide feedback to previous stages about how to correct the PIM and PSM models, as well as the transformation rules among them.

3. At the CM level it is possible to evaluate the final user interface (3 in Fig.1). The set of measurable attributes that can be evaluated at this level is the widest since more aspects related to the end-user perspective can be considered (i.e. browser compatibility, metaphor recognition, subjective appealing, etc). This evaluation will also generate a usability report in order to provide feedback to previous stages about how to correct the PIM and PSM models, as well as the transformation rules among them, and code generation rules among PSM and CM.

The former evaluations can be applied in an iterative way until the models (PIM, PSM, and CM) have the required level of usability. In order to integrate these evaluations into a framework, a usability evaluation process should be defined as an inspection method that guides evaluators on how the Web Usability Model can be applied. This inspection method could be defined in order to be compliant with the Quality Evaluation Division proposed in the ISO/IEC 2504n SQuaRE series [10]. The main steps to be included are:

1. Establish evaluation requirements such as the purpose of evaluation, identification of Web application type, and selection of the more relevant sub-characteristics of the Web Usability Model taking into account the users' needs.
2. Specify the evaluation concerning with the establishment of the artifacts to be evaluated (PIM, PSM or CM); selection of metrics associated to the attributes selected from the Web Usability Model; specification of the calculation formulas of these metrics taking into account the abstraction level of the artifact and the modeling primitives from the concrete Model-Driven Web development method; establishment of rating levels for these metrics; establishment of criteria for global assessment; and the definition of templates to report usability problems.
3. Design the evaluation plan describing the evaluator tasks schedule.
4. Execute the evaluation by applying the selected Web metrics in order to detect usability problems.
5. Generate the usability reports providing feedback in order to improve the intermediate artifacts (PIM and PSM) or transformation rules.
6. Analysis of changes suggested by usability reports and selection of the alternatives taking into account criteria such as level and priority of usability problems, resources needed to apply changes, etc.

It should be noted that this process is defined to be instantiated into any concrete Model-Driven Web Development method. The instantiation implies to know the modeling primitives of the concrete Model-Driven Web development method in order to be able to specify the calculation formula of the metrics and to understand the traceability between models. This traceability helps the evaluator to establish the source of the usability problems (PIMs, PSMs or transformations rules among them).

4. THE WEB USABILITY MODEL

The SQuaRE standard [10] proposes three different views for a quality model. These views are related to the context where the model will be applied: *Software Quality Model* to evaluate a concrete software product; *Data Quality Model* to evaluate the quality of the data managed in the product; and *Quality in Use Model* to evaluate how the stakeholders achieve their goals in a specific context of use.

Our Web Usability Model is an adaptation and extension from the usability model for model-driven development presented in Abrahão and Insfran [3], specifically, the model was adapted to be compliant with the *Software Quality Model* proposed in the SQuaRE.

The main quality characteristics of the software quality model are: *functionality*, *security*, *interoperability*, *reliability*, *operability (usability)* and *efficiency*. Although the term *operability* and *ease of use* have been proposed in SQuaRE to rename *usability* and *operability* sub-characteristic, respectively, we prefer to use the term *usability* and *operability* in this work to avoid misunderstandings in terminology.

Usability can be decomposed into the five sub-characteristics proposed in SQuaRE [10]: *learnability*, *understandability*, *ease of use (operability)*, *attractiveness* and *compliance*. The former three sub-characteristics are related to user performance and can be quantified mainly using objective measures. The last two sub-characteristics are related to the perception of the end-user or evaluator using the Web Application and can be quantified mainly using subjective measures.

The former three sub-characteristics were decomposed into other sub-characteristics or measurable attributes, taking into account the ergonomic criteria proposed in Bastien and Scapin [4]:

i. *Learnability* refers to the attributes of a Web application that facilitate learning: a) *help facilities* such as on-line

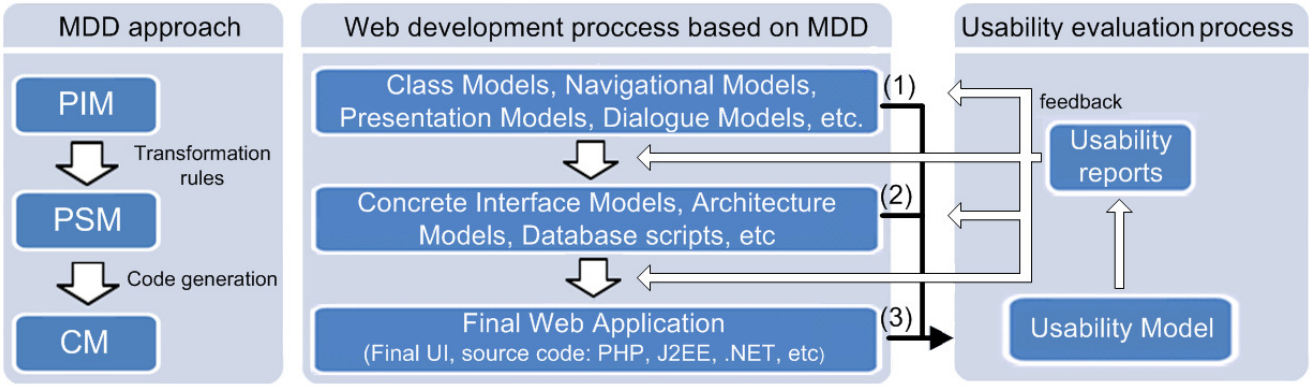

Fig. 1 Integrating a Usability Evaluation Process into a Model-Driven Web development process

help, contact section, etc; b) *predictability*, which refers to the ease with which a user can determine the result of his/her future actions (i.e. significance of link/image titles); c) *informative feedback* in response to user actions; and d) *memorability* as a measure of how quickly and accurately users can remember how to use a Web application that they have used before.

ii. *Understandability* refers to the attributes of a Web application that facilitate understanding: a) optical *legibility* of texts and images (e.g., font size, text contrast); b) *readability*, which involves aspects of information-grouping cohesiveness and density; c) *familiarity*, the ease with which a user recognizes the user interface components and views their interaction as natural; d) *brevity*, which is related to the reduction of user cognitive effort; and finally, e) *user guidance*, which is related to message quality, immediate feedback (to show the current user state), and navigability (to guide the user and to improve the access to the Web content).

iii. *Operability* refers to the attributes of a Web application that facilitate user control and operation: a) *execution facilities* such as compatible browsers, plug-ins needed, and update frequency; b) *data validity* of the user inputs; c) *controllability* of the services execution such as cancel, undo and redo support; d) *capability of adaptation* which refers to the capacity of the Web application to be adapted to the users' needs and preferences and e) *consistency* in the execution of services and control behavior.

The last two sub-characteristics are related to the perception of the end-user (*attractiveness*) or evaluator (*compliance*) using the Web Application:

iv. *Attractiveness* refers to the attributes of a Web application that are related to the aesthetic design. They can be quantified by measuring the UI uniformity in terms of *font style* (color, face and size), *background color*, and *position of elements*.

v. *Compliance* can be measured by assessing the agreement of the proposed Web Usability Model with respect to the standard SQuaRE and several Web design style guides.

Once the sub-characteristics have been identified, Web metrics are associated to the measurable attributes in order to quantify them. Values obtained from these Web metrics will allow us to interpret if measurable attributes contribute to achieving certain usability level in the Web application. The metrics included in our model were mainly extracted from the survey presented in Calero *et al.* [5]. We only selected those metrics that were theoretically and/or empirically validated. In addition, we proposed new metrics for several measurable attributes that were not appropriated covered by this survey.

As an example, we show some definitions of new proposed metrics that can be associated to attributes of the Web Usability Model:

- *Number of different font styles for textual links:* This metric is defined as number of different font style combinations (size, face, and color) for all textual links in the same navigation category. (Scale type: absolute value greater or equal to 1). The interpretation is: more than one style combination in the same navigation category means that font style uniformity is not insured. This metric is associated to the *font style uniformity* attribute, which belongs to the *attractiveness* sub-characteristic (iv).

- *Proportion of elements that show current user state*: This metric is defined as the ratio between the number of elements that show feedback about the current user state and the total number of elements that are required to have this feedback capability. (Scale type: ratio between 0 and 1). The interpretation is: values closer to 1 indicate that user can obtain feedback about his/her current state in the Web application. This metric is associated to the *immediate feedback* attribute, which belongs to the *user guidance* sub-characteristic (ii. e).

Web metrics definitions from the Web Usability Model are generic, and their calculation formula must be instantiated by identifying variables from this formula in the modeling primitives of the concrete Web development method for each abstraction level (PIM, PSM or CM). Not all the metrics can be defined at all the abstraction levels, for instance, the former metric can be applied at PIM level (if style properties are defined at the abstract UI) or at CM level (if style properties are defined in Cascading Style Sheets files). However, the second metric only can be defined at PSM or CM level since the feedback depends on the widget behavior from the concrete interface.

5. INSTANTIATION OF THE USABILITY EVALUATION PROCESS

In this section, we show an overview of how the previous usability process can be instantiated into a concrete Web development methodology. As an example, we selected the OO-H [7] method.

The OO-H method [7] provides designers with the semantics and notation for developing Web applications. The set of conceptual models that represents the different concerns of a Web application are: the specification of content requirements (Class Model) and the specification of functional requirements in terms of navigation needs (Navigation Model, NAD). A merge between the class and navigation models results in an Abstract Presentation Diagram as an integrated PIM model, which presents an abstract user interface as a collection of abstract pages. APD can be refined by a pattern catalog. Finally, platform-specific models (PSMs) are automatically obtained from the APD, from which source code (CM) can be automatically generated.

Next, we show as an example, a brief description about the steps involved in our usability evaluation process.

Step 1 (See Section 3): The purpose is to evaluate the usability of a Web application developed following the OO-H method. The selected Web application is a task management system developed for a Web development company located in Alicante, Spain. Finally, the attributes chosen were *font style uniformity* to evaluate the *attractiveness* sub-characteristic, and *immediate feedback* to evaluate the *user guidance* sub-characteristic, at least to some extent.

Step 2 (See Section 3): The artifacts selected for this evaluation were the final UIs (Code Model). The metrics selected to evaluate the previous attributes were *Number of different font styles for text links* and *Proportion of elements that show current user state* (see explanation of each metric in Section 4). The rating level for the former metric was established at *no UP* for values equal to 1; *low UP* for values equal to 2; *medium UP* for values equal to 3; and *critical UP* for values greater than 3. The rating level for the second metric was established at *no UP* for values equal to 1; *low UP* for values in the range [0.8, 1};

medium UP for values [0.5, 0.8} and *critical UP* for values [0, 0.5}. The usability report is defined as a list of usability problems (UP) detected with the next fields: description of the UP, level of the UP (critical, medium, or low), source of the problem (model), occurrences, and recommendations to correct it. More fields can be defined such as priority, impact, etc.

Step 3 (See Section 3): In this case, the evaluator is the same developer. The task assigned was the evaluation of all the user interfaces (CM) in order to present a usability report which will contain the usability problems detected with all the proposed fields filled in.

Step 4 (See Section 3): As an example, we only show the execution of the evaluation of one user interface (CM). Figure 2 shows a user interface automatically generated (Code Model) that represents the task management functionality of the Web application.

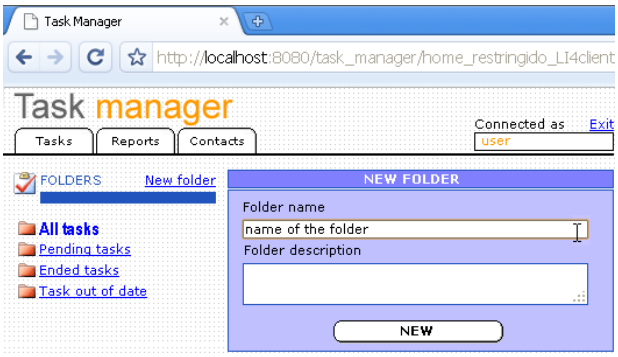

Fig.2 Example of a User interface automatically generated (Code Model)

The selected metrics were applied:

1. *Number of different font styles for textual links* [1]: The textual links that appears in the user interface (Fig. 2) are *Tasks*, *Reports*, *Contacts* and *Exit* from the top menu; and *New Folder*, *All tasks*, *Pending tasks*, *Ended tasks*, and *Tasks out of date* from the left menu. In the first navigation category (top menu), the value of the metric is 2 since the links *Tasks*, *Reports*, *Contacts* are displayed in a different style from the *Exit link*, which is displayed in a different color and it is also underlined. In the second navigation category (left menu), the value of the metric is also 2 since the links *New Folder*, *Pending tasks*, *Ended tasks*, and *Tasks out of date Contacts* are displayed in a different style from the *All tasks*, which is displayed in a different font face and font size. The rating level of the metric (see Step 2) indicates the existence of a low usability problem (UP001) for each menu.
2. *Proportion of elements that show current user state* [1]: The user interface must show the current user state, it means, the current section and the current task that is being performed. There are three types of elements that show the current user state in the Web application: the tabs from the top menu (*Tasks*, *Reports,* and *Contacts)*; the shape changes of the cursor when it is pointing on a textbox; and the highlight effects of a textbox when it has focus. Since the tabs are the only type of element that does not explicitly show the section in which the user is currently interacting, the value of the metric is 2/3=0.66. The rating level of the metric (see Step 2) indicates the existence of a medium usability problem (UP002).

Steps 5 and 6 (See Section 3): The usability problems detected after applying the previous metrics, can be explained in a usability report that contains the UP001 (See Table 1) and the UP002 (See Table 2).

Table 1. Usability problem detected: UP001

id	UP001
Description	The links *Tasks*, *Reports*, and *Contacts* are displayed in a font style that is different from the font style of the *Exit* link. The same problem occurs with the *all tasks* link that is displayed in a font style that is different to the used in the links: *New Folder*, *Pending tasks*, *Ended tasks*, and *Tasks out of date.*
Affected attribute	Attractiveness / font style uniformity.
Level	Low (rating level: 2).
Source of the problem	Abstract Presentation Diagram (PIM model).
Occurrences	2 occurrences (top menu and left menu)
Recommendations	Change the font style properties for the links *Tasks*, *Reports*, *Contacts* and *all tasks* in the Abstract Presentation Diagram. In this PIM model font style properties can be defined.

Table 2. Usability problem detected: UP002

id	UP002
Description	Tabs do not show the current user state in the Web application.
Affected attribute	Understandability/ User Guidance/ Immediate feedback.
Level	Medium (rating level: 0.66)
Source of the problem	The transformation rule that maps the representation of the tabs: *Task, Reports* and *Contacts* (PIM level) with the specific widget of the platform that shows the tabs (PSM).
Occurrences	1 occurrence for each UI that shows these tabs.
Recommendations	The widget target of the transformation rule should be changed for other widget with a highlight feature when a tab is clicked.

After analyzing and applying the proposed recommendations, a more usable Web application can be obtained without to need maintenance of source code.

6. DISCUSSIONS AND FUTHER WORK

This paper has presented a proposal in progress to integrate a usability evaluation process into Model-Driven Web development processes. The purpose of our work is to give an outline of a generic usability evaluation process supported by a Web Usability Model. A preliminary version of a usability evaluation process supported by a Web usability Model has been presented. Our Web Usability Model decomposes the

[1] It should be note that both metrics must be applied to all the user interfaces of the entire Web application.

usability sub-characteristics (from the Software Quality Model proposed in the ISO/IEC 25000 SQuaRE standard) into other sub-characteristics and measurable attributes taking into account ergonomic criteria. Web metrics were associated to measurable attributes in order to quantify them. Finally, a brief example has been shown in order to illustrate how the usability evaluation process can be instantiated into a concrete Web development method that follows the MDD approach. Although our example only shows a CM evaluation providing feedback to PIM models or transformations between PIM and PSM models, the usability evaluation process can evaluate intermediate artifacts (PIM and PSM models) by selecting metrics that their calculation formula has been defined to be applied to concrete PIM and PSM models (i.e., depth and breadth of a navigational map [1] associated to the navigability attribute).

We believe that the inherent features of model-driven development processes (e.g., traceability between models by means of model transformations) provide a suitable environment for performing usability evaluations. Specifically, if the usability of an automatically generated user interface can be assessed, the usability of any future user interface produced by this approach could be predicted. In other words, we are talking about a user interface that is usable by construction [2], at least to some extent.

In this way, usability can be taken into account throughout the entire Web development process, enabling Web applications to be developed with better quality thereby reducing effort at the maintenance stage.

Further work is intended to:

- Perform an entire instantiation of the usability evaluation process into the OO-H method.
- Define guidelines in order to guide evaluators on how the Web Usability Model can be applied
- Explore aggregation mechanisms for aggregating values obtained by individual metrics, and perform analyses of the impact on how the attributes affect (negatively or positively) other attributes of the Web Usability Model.
- Instantiate the evaluation process into different Model-Driven Web Development methods in order to improve our approach.
- Develop a tool to support the entire usability evaluation process. The tool will be able to manage the Web Usability Model by creating a repository of catalogued metrics following the SQuaRE patterns.

7. ACKNOWLEDGMENTS

This work is financed by META project (ref. TIN2006-15175-C05-05), the Quality-driven Model Transformation Project from the Universidad Politécnica de Valencia. The authors thank Jaime Gomez from Universidad de Alicante for his valuable help in providing the generated Web application and its models used to illustrate our usability evaluation process.

8. REFERENCES

[1] Abrahão, S., Condori-Fernández, N., Olsina, L., and Pastor, O. 2003. Defining and Validating Metrics for Navigational Models. Proc. of the 9th Inter. IEEE Software Metrics Symposium, 200-210.

[2] Abrahão, S., Iborra, E., and Vanderdonckt J. 2007. Usability Evaluation of User Interfaces Generated with a Model-Driven Architecture Tool. Maturing Usability. Springer HCI series, Vol. 10, 3-32.

[3] Abrahão, S. and Insfran, E. 2006. Early Usability Evaluation in Model-Driven Architecture Environments. Proc. of the 6th IEEE International Conference on Quality Software. IEEE Computer Society, 287-294.

[4] Bastien, J.M. and Scapin, D.L. 1993. Ergonomic criteria for the evaluation of human-computer interfaces. Tech. Rep. num.156. INRIA, Rocquencourt, France.

[5] Calero C., Ruiz J., and Piattini M. 2005. Classifying Web metrics using the Web quality model. Emerald Group Publishing Limited. Vol. 29(3), 227-248.

[6] Ceri, S., Fraternali, P., and Bongio, A. 2000. Web Modeling Language (WebML): A Modeling Language for Designing Web Sites. Proc. of the 9th WWW Conference, 137-157.

[7] Gómez, J., Cachero, C., and Pastor, O. 2001. Conceptual Modeling of Device-Independent Web Applications. IEEE MultiMedia, Vol. 8(2), 26-39.

[8] ISO/IEC. 1998. ISO/IEC 9241-11, Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 11: Guidance on Usability.

[9] ISO/IEC. 2001. ISO/IEC 9126-1 Standard, Software Engineering, Product Quality - Part 1: Quality Model.

[10] ISO/IEC. 2005. ISO/IEC 25000 series, Software Engineering, Software Product Quality Requirements and Evaluation (SQuaRE).

[11] Kraus, A., Knapp, A., and Koch, N. 2006. Model-Driven Generation of Web Applications in UWE. 3rd Inter. Workshop on Model-Driven Web Engineering.

[12] Moraga, M.A, Calero, C., Piattini, M., and Diaz, O. 2007. Improving a portlet usability model. Software Quality Control, Vol. 15(2), 155-177.

[13] Olsina, L. and Rossi, G. 2002. Measuring Web Application Quality with WebQEM. IEEE Multimedia, Vol. 9(4), 20-29.

[14] Panach, J., Condori-Fernández, N., Valverde, F., Aquino, N., and Pastor, O. 2007. Towards an Early Usability Evaluation for Web Applications. International Conference on Software Process and Product Measurement. LNCS Springer Vol. 4895, 32-45.

[15] Seffah, A., Donyaee, M., Kline, R.B., and Padda, H.K. 2006. Usability Measurement and Metrics: A Consolidated Model. Software Quality Journal, Vol. 14(2), 159-178.

[16] Ivory, M.Y. 2001. An Empirical Foundation for Automated Web Interface Evaluation. PhD Thesis. University of California, Berkeley, Computer Science Division.